Breast Cancer Dataset KNN and LDA Analysis

In the below step I load the Kaggle data taken from https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data and split our data into a training set and a test set with 80% in training set and 20% in test set.

```
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
set.seed(123)

data <- read.csv("data.csv")
train_idx <- sample(1:nrow(data),
size = round(0.8*nrow(data)),
                    replace = FALSE)
train_set <- data[train_idx,]
test_set <- data[-train_idx,]
```

Then I define a training control and pick 5 candidate models with k = 1, 3, 5,7 and 9 and then pick the model with the best-tuned value of folds k.

```
#Define the training control
train_control <- trainControl(method = "cv", number = 10)

# cross-validate the knn model with candidate ks
knn_model <- train(diagnosis ~radius_mean + texture_mean +
                     perimeter_mean + area_mean + smoothness_mean +
                     compactness_mean + concavity_mean + concave.points_mean +
                     symmetry_mean + fractal_dimension_mean + radius_se + texture_se +
                     perimeter_se + area_se + smoothness_se + compactness_se +
                     concavity_se + concave.points_se + symmetry_se + fractal_dimension_se +
                     radius_worst + texture_worst + perimeter_worst + area_worst +
                     smoothness_worst + compactness_worst + concavity_worst +
                     concave.points_worst + symmetry_worst + fractal_dimension_worst,
                   data = train_set,
                   method = "knn",
                   tuneGrid = data.frame(k = c(1,3,5,7,9)),
                   trControl = train_control,
                   preProcess = c("center","scale"))

print(knn_model$bestTune)
```

```
##   k
## 4 7
```

```
print(knn_model$results)
```

```
##   k  Accuracy     Kappa AccuracySD    KappaSD
## 1 1 0.9494203 0.8876221 0.02076864 0.04622724
```

```
## 2 3 0.9647826 0.9208179 0.01863013 0.04218461
## 3 5 0.9648309 0.9211837 0.02594222 0.05832252
## 4 7 0.9648792 0.9216522 0.02791831 0.06156360
## 5 9 0.9647826 0.9206675 0.02148350 0.04832184
```

As shown above we fet the highest accuracy for k =7 at 0.9648. We can also compare this with the LDA model and see if that is more accurate than KNN.

```
#print(knn_model$results)
```

```
lda_model <- train(diagnosis ~radius_mean + texture_mean +
                     perimeter_mean + area_mean + smoothness_mean +
                     compactness_mean + concavity_mean + concave.points_mean +
                     symmetry_mean + fractal_dimension_mean + radius_se + texture_se +
                     perimeter_se + area_se + smoothness_se + compactness_se +
                     concavity_se + concave.points_se + symmetry_se + fractal_dimension_se +
                     radius_worst + texture_worst + perimeter_worst + area_worst +
                     smoothness_worst + compactness_worst + concavity_worst +
                     concave.points_worst + symmetry_worst + fractal_dimension_worst,
                   data = train_set,
                   method = "lda",
                   trControl = train_control)
print(lda_model$results)
```

```
##   parameter  Accuracy     Kappa AccuracySD    KappaSD
## 1      none 0.9559354 0.8996049 0.02747882 0.06333826
```

As seen above, the LDA model is not as accurate with its accuracy at 0.956.

```
fitted_lda <- train(diagnosis ~ radius_mean + texture_mean +
                     perimeter_mean + area_mean + smoothness_mean +
                     compactness_mean + concavity_mean + concave.points_mean +
                     symmetry_mean + fractal_dimension_mean + radius_se + texture_se +
                     perimeter_se + area_se + smoothness_se + compactness_se +
                     concavity_se + concave.points_se + symmetry_se + fractal_dimension_se +
                     radius_worst + texture_worst + perimeter_worst + area_worst +
                     smoothness_worst + compactness_worst + concavity_worst +
                     concave.points_worst + symmetry_worst + fractal_dimension_worst,
                    data = train_set,
                    method = "lda",
                    preProcess = c("center","scale"),
                    trControl = trainControl(method = "none"))
```

```
cf <- confusionMatrix(predict(fitted_lda,test_set), as.factor(test_set$diagnosis))
print(cf)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  B  M
##          B 60  4
##          M  1 49
##
##                Accuracy : 0.9561
##                  95% CI : (0.9006, 0.9856)
##     No Information Rate : 0.5351
##     P-Value [Acc > NIR] : <2e-16
```

```
##
##                   Kappa : 0.9115
##
##  Mcnemar's Test P-Value : 0.3711
##
##             Sensitivity : 0.9836
##             Specificity : 0.9245
##          Pos Pred Value : 0.9375
##          Neg Pred Value : 0.9800
##              Prevalence : 0.5351
##          Detection Rate : 0.5263
##    Detection Prevalence : 0.5614
##       Balanced Accuracy : 0.9541
##
##        'Positive' Class : B
##
```